CollECTeR 2007, 9-11 December, Melbourne Australia

Designing Secure e-Contracting Systems[1]

Praveen Gauravaram and Ernest Foo
Information Security Institute
Queensland University of Technology
Australia
(p.gauravaram@isi.qut.edu.au|e.foo@qut.edu.au)

Abstract

Information security goals and services for e-contracting systems have not previously been closely examined in open literature. This paper identifies key security goals and the services to be considered when designing e-contracting systems. Several AEC collaborative platforms are reviewed. These systems are currently used for project management and could be used for e-contracting. The results of the review show that while integrity and confidentiality of contract documents is ensured during communication in contract formation, few of the systems consider these security requirements during archiving. A new e-contracting architecture is proposed to provide security for the full e-contract life cycle.

1. Introduction

Modern Architecture, Engineering and Construction (AEC) enterprises conduct business under a contract which constitutes a legally binding agreement enforced by the law between the enterprise and its customers or suppliers [20]. The automation of the contracting process, especially with the advent of the Internet in the past decade, has benefited these enterprises in improved productivity and security, effective aggregated contract information, speed-up of the contract life-cycle process, reduction in contractual errors and risk, profit optimization and better compliance [22].

A typical e-contracting system is a web-based software application used to conduct contracting between business parties where the e-contracting system works as a representation of paper contracts. Online collaborative platforms such as Aconex [4], Constructware [10], Causeway [11], are emerging as the principal systems for document management in the AEC sector. Such systems could potentially be used for e-contracting. However, there are currently no specific tools designed for e-contracting. The increasing use of the Internet as an effective business tool for e-contracting has motivated us to investigate the information security consequences that may flow from using the Internet as a medium to form contractual relationships. This paper identifies the security requirements and services of e-contracting systems.

In a closely related work, Knorr and Röhrig [21] have presented an open framework for the analysis of security requirements of business processes in electronic commerce. They have identified confidentiality, integrity, availability and accountability as the security objectives to suit the legal needs of a business process. Meier and Röhrig [23] have used a goal-oriented approach to derive appropriate security safeguards for the different contract types and discuss the implementation issues for electronic agent-

[1] The research described in this paper was sponsored by the Australian Cooperative Research Centre for Construction Innovation Project No. 2005-025-A.

based contracting systems. Both Knorr and Röhrig and Meier and Röhrig concentrate on contract formation. In this paper we consider security during contract formation, management and archiving.

This paper contributes by identifying the information security requirements that e-contracting systems must achieve. In addition, the paper presents results of an informal survey of several AEC collaborative platforms that could be used for e-contracting. Finally, the paper proposes a conceptual architecture to support non-repudiation, authenticity, integrity and proof of agreement throughout the e-contracting life-cycle. The paper is organised as follows: Section 2 defines the stages of an e-contracting process. Section 3.1 describes the e-contracting security goals. Section 3.2 describes security services essential for e-contracting. Section 3.3 provides the results of our survey on AEC collaborative platforms used for e-contracting. Section 4 proposes an architecture for the support of e-contracting life cycle which is discussed and informally analysed in section 4.4 of the paper. Finally, Section 5 concludes the paper.

2. E-contracting Basics

An electronic contract (e-contract) is an agreement created and signed in electronic form without using any paper or other hard copies [29]. It is a contractual agreement, represented as digital information and signed with the electronic or digital signatures of the participating parties [7].

The literature on the security of e-contracting concentrates on the process of e-contract formation. Some of the different methods that may be used to execute e-contracting include:

- Exchange of text documents using electronic communications such as email where a party which issues the contract, writes the contract on his/her computer and emails it to the second party by typing the name and the second party emails it back to the first party with the name indicating the acceptance of the contract. A name typed in the email is considered as an electronic signature [15]. While electronic signatures describe signatures incorporated in a document by cryptographic or non-cryptographic means, digital signatures specifically describe signatures based on cryptographic techniques.
- The text documents that form the basis of an e-contract may be written in XML, a mark-up language for documents containing structured information [32]. Structured information contains both content and some indication of what role that content plays. One advantage of forming contracts using XML is that contracts can be processed using machines and contracts can be imported into contract management and negotiation tools and achieving better specifications of the contract using industry specific XML vocabularies.
- An e-contract may be in the form of a “click to agree” contract. The terms and conditions of the contract are displayed on one party's website and the other party agrees to the contract by clicking an “I agree” button on the website accepting the relevant terms and conditions.
- Conceptual system architectures support either part of or the complete e-contracting life cycle process. For example, the multiple signing using digital signatures and dynamic update conceptual architectures described by Angelov et al [2] to manage the e-contract updates.

Generally, when contracting parties decide upon a particular method to execute e-contract life-cycle, their decision is influenced by the nature and importance of the relevant e-contract. For e-contracts of strategic importance or of high economic value, parties may wish to utilize appropriate mechanisms to achieve the security of relevant documentation.

An e-contracting process or life cycle consists of a number of phases where each phase constitutes activities confined to that phase. At a broad level, we classify e-contracting processes into three phases:

E-contract formation. The following steps take place during the formation of e-contracts.

1. Information. General contract preparations are made, information for a request or offer of services is provided and contracting parties are identified.
2. Pre-contract. Preparatory contracting process is performed where several contract negotiations are administered and managed.
3. Contract Negotiation. Contract negotiations are performed, preliminary agreements are made regulating the steps on the proceedings of negotiations and a draft agreement serving as an example of the final contract is established.

4. Enactment. The contract is finalised and work can commence or goods shipped. The enactment process is executed with signatures of all the participating parties.

E-contract management. Contracting parties may have to apply changes to the e-contracts signed in the enactment phase [2]. The e-contracting management system incorporates any variations (updates) of the e-contracts formed among the contracting parties.

E-contract archiving. The finalised e-contracts along with other related contractual communications among the business parties are archived for future evidential purposes.

3 Security in E-Contracting

In this section the security goals that should be achieved for secure e-contracting are proposed. We then describe the general security and cryptographic mechanisms that can provide these goals. The use of these security mechanisms is reviewed in web-based AEC collaborative platforms.

3.1 E-contracting Security Goals

Any system used to perform the e-contracting process must ensure that the whole process is secure. The security requirements specifically for the e-contracting process have not been closely examined in the literature to date though the security requirements of other business processes such as e-commerce and e-business have been explored [21, 27]. The identified security requirements have been determined based on standard e-commerce security requirements, a survey of existing AEC collaborative platforms and interviews with contracting staff in the AEC industry.

The security goals of an e-contracting system are outlined below:

Confidentiality ensures protection of e-contracts and other communications from unauthorized disclosure in the e-contracting system. The contracting parties may not want to disclose the documents in every stage of the e-contracting life cycle to unauthorized parties. This condition depends on the agreement undertaken among the contracting parties before the start of the e-contracting process and on the type of business undertaken by the parties.

Integrity ensures that contractual documents exchanged among the contracting parties or stored in the e-contracting system at any point of time are not duplicated, modified or deleted. The contracting parties aim for secure storage or transmission of all the contracting documents and messages in every stage of the e-contracting life cycle.

Authenticity ensures that the parties involved in e-contracting are exactly who they claim to be. The contracting partners must authenticate themselves to the e-contracting system and their credentials need to be recorded and maintained throughout the period of e-contracting.

Non-repudiation ensures that contracting parties are prevented from denying having performed actions such as denying an established contract and denial of sending or receiving any messages.

Availability ensures that e-contracting systems and contractual data are available to the authorized personnel during the period of contract life cycle.

Proof of agreement ensures every action of the contracting parties in the e-contracting system throughout the period of its usage is considered as a proof to which they have agreed.

Proof of existence assures the existence of contractual documents in the e-contracting system or documents communicated via the Internet at a point of time.

3.2 Security Services of E-contracting Systems

An e-contracting system should incorporate mechanisms to ensure that contractual evidence is securely gathered and stored in the case of disputes among the contracting parties. These mechanisms are the security services that provide the security goals defined in Section 3.1.

3.2.1 Secure Access Control: To alleviate concerns about the security of e-contracts and the messages communicated, e-contracting systems must be designed so that users have limited access to the e-contracts, depending on their role within the enterprise or business. For example, in a collaborative platform used for e-contracting only a sub-contractor may be allowed to access drawings and communications relating to a particular project while other project information is shared with other parties involved with the enterprise [5].

The rights to access, view, modify or delete contractual data in an e-contracting system are controlled by an access control system which the e-contracting system supports. The components of an access control system are:

User-authentication. User authentication to the e-contracting system is a process of verifying the identity of the user to the system where the user confirms to the system who he or she is. E-contracting systems with only password based authentication provide sufficient level of security.

Authorization. Authorization refers to the permissions or rights of the user to read, write (for example, add, create, delete or rename the e-contract files in the system) and execute contractual data in the e-contracting system. A security policy determines who will have access to different types of contractual information and whether or not they have a right to alter the data. The method by which the security policy is implemented is referred to as a security model [17].

3.2.2 Secure Communication: The collaborating parties need to address the effect of the risk involved in the exchange of e-contracts on their confidentiality and integrity in transmission. If the information and communications technology (ICT) used for communications in the e-contracting process is secure, the e-contracting process itself is

partially secured. This is also the case for most e-commerce applications. The personnel who use e-contracting systems need to make sure that these systems use secure Internet protocols such as secure sockets layer (SSL) [16] or Transport Layer Security (TLS) [12] to provide confidentiality, integrity and authenticity to the data in transmission.

3.2.3 Secure Recording and Archiving: Upon the completion of e-contracting, all the contractual documents processed in different stages of the e-contracting life cycle need to be archived for future evidential purposes. Secure archiving of contractual information requires durability of the storage media and readability of contractual documents. Generally, contracting parties will not be found to have satisfied their obligation to preserve contractual documents if the mechanism on which they are stored has broken down or if the documents are saved in a format that is no longer able to be read by contemporary computer systems. E-contracting systems need to be updated regularly so that the contracting parties do not have to be concerned with the continued readability and availability of the documents [33]. The parties should agree contractually before contracting, how the contracting data will be archived and what data will remain available to each project participant [5].

The main mechanisms for a secure archiving service are as follows:

Digital Signatures and Hash Functions. Digital signatures [26] based on the combination of public key cryptography (PKC) and cryptographic hash functions ensure data origin authentication, integrity of the signed contract, non-repudiation and proof of agreement. Digital signatures do not bind contracts to a particular time of origin. Usage of secure hash functions such as SHA-256 [1] and the secure use of PKC ensure secure digital signatures. The security properties of hash functions are outlined below:

- Collision resistance [24]: For a hash function H, it should be hard to find two messages M_1 and M_2 such that $M_1 \neq M_2$ and $H(M_1) = H(M_2)$.
- Preimage resistance [24]: Given the hash value $Y = H(M)$ for a hash function H, it must be hard to find M.
- 2nd-preimage resistance [24]: Given a message M_1, it must be hard to find another message M_2 such that $M_2 \neq M_1$ and $H(M_1) = H(M_2)$.

Logging and Auditing. For evidential reasons, often specific tasks need to be recorded. System event logs can be used to provide proof of existence of documents and events such as the application of a digital signature to a contract document. Although admissible as evidence, system logs do depend on the trustworthiness of the system which records the event. A trusted third party such as a time-stamping authority may be more reliable source of evidence.

Digital Time-stamping. Time-stamping of digital signatures by a time-stamping authority (TSA) on contractual documents and archived documents provides proof of existence of those documents at a given point of time [28]. As it is assumed that the TSA is an impartial fair third party, it can later be undoubtedly demonstrated that digital signatures on the contract have been valid at the time of time-stamping. The accuracy of a time stamp depends on the accuracy of the timeserver that allows the TSA to synchronise its system clock over the Internet. The time information provided by the timeserver to a TSA is directly traceable to the Universal Time Code. Accuracies, for example, of 1-50 milliseconds can be achieved using Network Time Protocol (NTP version 3) depending on the characteristics of the synchronization source [25]. The security properties expected from the time stamp issued by a TSA are outlined below:

- It must be infeasible for a time stamping authority to time stamp a document with a date and time that is different from the correct one.
- It must be infeasible to change even a single bit of a time stamped document without the change being apparent.
- Relative temporal authentication [6, 19, 18] intuitively combines message authentication with the notion of timeliness of messages. The TSA is said to provide this property if one is able to decide which stamp has been issued first for each pair of time stamp. This is achieved by applying a collision resistant hash function to the earlier stamps which is then incorporated in the later time stamp.
- The TSA must be reliable and available when needed [3].

3.3 E-contracting in Collaborative Platforms

Web-based AEC collaborative platforms designed for project collaboration among companies can be used as the basis to conduct e-contracting. The on-line collaboration platforms given in Table 1 have been reviewed based on the vendor claims and the review of the content in the documentation available in their websites. In the Table 1, "Y" stands for Yes, "T" for Transmission and "?" for Unknown.

Table 1. Security Services of Reviewed Collaborative Platforms

Platform	Authentication	Access Control	Integrity	Confidentiality	Log and Auditing
Aconex [4]	Y	Y	Y	T	Y
Citadon CW [8,9]	Y	Y	T	Y	Y
Contructware [10]	Y	Y	T	T	Y
TeamBinder[14]	Y	Y	T	T	?
ECM [11]	Y	Y	?	?	Y
E-Builder [13]	Y	Y	?	?	Y
Evoco [32]	Y	Y	?	?	Y
Information Channel [31]	Y	Y	?	?	Y

Many of these platforms achieve the security goals required for e-contracting without the need of all the security mechanisms described in section 3.2. Secure e-contracting can be facilitated by providing user authentication and access control to achieve document integrity and confidentiality and logging and audit mechanisms to achieve non-repudiation, proof of agreement and proof of existence. The results in Table 1 highlights that all of the reviewed construction collaboration platforms provide user authentication through a username and password mechanism. Every platform provides role-based access control to restrict the availability of documents to only the authorised personnel. Most of the reviewed applications

log user actions and have a file version control mechanism where a new file version is created for every update to a contract document.

Many platforms provide document integrity and confidentiality during the transmission and uploading of the documents. Very few of the applications provide document integrity and confidentiality after a project has finished. We address this issue by proposing a new e-contracting architecture.

4 An E-contracting Architecture

We propose a two-party e-contracting system architecture which uses cryptographic tools to ensure that the security goals defined in section 3.1 are achieved for e-contracting documents during contract formation, management and archiving. It is assumed that all messages are transmitted over a channel secured by TLS or similar mechanism which provides confidentiality between the two parties. The main emphasis of the security is on providing integrity, authenticity and proof of agreement.

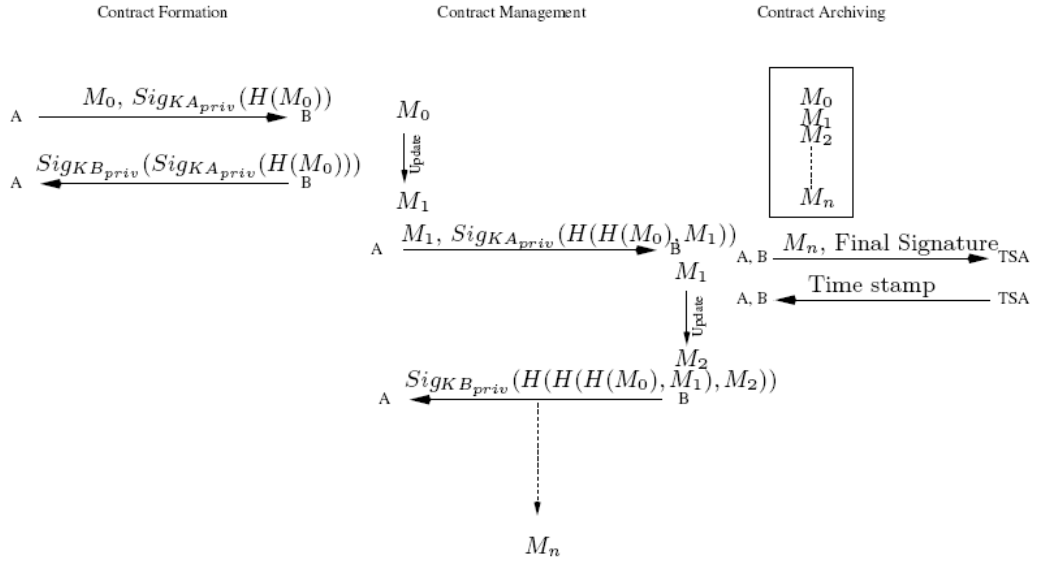


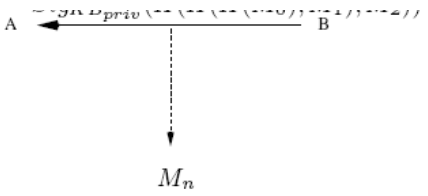
Figure 1. An E-contracting Architecture for Contract Formation, Management, and Archiving

This architecture consists of three individual systems meant for e-contract formation, e-contract management and e-contract archiving. These systems are summarised in Figure 1 assuming the participation of two contracting parties A and B.

4.1 E-contract Formation

There are many methods for contract formation. To ensure authenticity and non-repudiation, this system uses digital signatures to form the electronic contract.

1. While enacting an e-contract M_0, party A computes the hash value $H(M_0)$ of M_0 using the cryptographic hash function H and signs the hash value $H(M_0)$ using his/her private key KA_{priv} and the signature is $Sig_{KApriv}(H(M_0))$. Party A sends the message M_0 and its signature $Sig_{KApriv}(H(M_0))$ to B.



2. Party B first verifies the signature of A using the public key KA_{pub} of A ensuring that A has signed the message M and its integrity has been maintained. Then B signs the signature of A using his/her private key KB_{priv}. The party B then sends the signature $Sig_{KBpriv}\ (Sig_{KApriv}\ (H(M_0)))$ to A.
3. A verifies the signature $Sig_{KBpriv}\ (Sig_{KApriv}\ (H(M_0)))$ using the public key KB_{pub} of B. During the execution of all the above steps, both parties A and B store the messages, their hash values and signatures.

4.2 E-contract Management

This system manages variations to the e-contract. The following steps take place during contract variation.

1. The party, for example A, which wants to update the contractual document M_0 enacted in the previous stage, updates it to M_1.
2. Then A signs M_1 by concatenating the updated contractual document M_1 to the digest $H(M_0)$ of the previous contractual document M_0. This compound document is represented as $(H(M_0),M_1)$ where , represents the concatenation operation.
3. The party A then computes the digest $H(H(M_0),M_1)$ of the compound document $(H(M_0),M_1)$.
4. The party A then signs the digest $H(H(M_0),M_1)$ using its private key KA_{priv} and sends the signature $Sig_{KApriv}\ (H(H(M_0),M_1))$ and the updated contract M_1 to B.
5. The party B then verifies the signature on M_1 and if needed updates M_1 to M_2 and signs $H(H(H(M_0),M_1),M_2)$.
6. The party B then sends M_2 along with the signature $Sig_{KBpriv}\ (H(H(H(M_0),M_1),M_2))$ to A who verifies this signature.
7. Party A again updates the document M_2 if needed and this process of updating continues until both the parties A and B agree on the final contractual document M_n as shown in Fig 1.
8. Both parties sign on the final contract M_n by linking it to the previously established hash chain $H(...H(H(H(M_0),M_1),M_2),...,M_{n-1})$. The final digest formed with the hash chain is:

$$H(...H(H(H(H(M_0),M_1),M_2),...,M_{n-1}),M_n)$$

Assuming that it is computed by the party A, the final signature is:

$$Sig_{KApriv}\ (Sig_{KBpriv}\ (H(...H(H(H(H(M_0),M_1),M_2),...,M_{n-1}),M_n)))$$

4.3 E-contract Archiving

This system prepares the contract documents for long term storage and aims to ensure a chain of custody using the hash chain and signature mechanisms. Both parties securely archive all the contractual documents $M_0,M_1,...,M_n$, their digests and all signatures. The signatures of both the parties on the final hash chain are time stamped. In general, it is not necessary to time stamp all the contractual documents processed in the e-contracting life

cycle as the time of updating is recorded in a local log in the system. We assume that the TSA used for this purpose is a trusted third party authority.

The following protocol is used to time stamp the final signature.

1. Both parties A and B submit their final signature Sig_{KApriv} $(Sig_{KBpriv}(H(\ldots H(H(H(H(M_0),M_1),M_2),\ldots,M_{n-1}),M_n)))$ on the hash chain to the TSA.
2. This signature on the final hash chain is verified by the TSA using the respective public keys KA_{pub} and KB_{pub} of A and B.
3. The TSA time stamps the signature *Final Signature* = Sig_{KApriv} $(Sig_{KApub}$ $(H(\ldots H(H(H(H(M_0),M_1),M_2),\ldots,M_{n-1}),M_n)))$ by attaching the date and time to the signature and signing the compound document with its private key $K\text{-}TSA_{priv}$. Let this time stamp be $SigK\text{-}TSA_{priv}$ $(Final\ Signature||time||date)$.
4. The TSA then sends the time stamp to both the clients who verify the time stamp using the public key $K\text{-}TSA_{pub}$ of the TSA.

The TSA signing process must be repeated at regular time intervals specified in the contract agreement. This process is conducted using the latest hash functions and signatures to ensure that the long term integrity of the documents is achieved. This process is necessary as it is assumed that contract documents can be stored for several years and that new stronger algorithms will be developed that will supersede current algorithms. Older signatures and hash chains will need to be encapsulated with the contract documents and retained as proof that documents have not been altered when the signatures are renewed.

This architecture can be easily extended to the case of e-contracting for more than two parties. In this case, contracts are signed by all the parties one after the other during the e-contract formation step. Assuming the existence of three participants A, B and C, the contract signed by A is verified and signed by B which is then verified and signed by C. Finally C sends the signed contract to both A and B who verify the contract. Similarly, the e-contract update management system can be designed for multi-party scenario. We consider the further design issues and analysis of this e-contracting architecture for multi party scenario as part of the future work.

4.4 Analysis of the proposed E-contracting Architecture

The implementation of public key infrastructure (PKI) and secure cryptographic hash functions to sign the contractual documents and the messages communicated across the Internet in the above e-contracting system architecture provides non-repudiation, authenticity, integrity and proof of agreement security goals. The e-contracting system uses a hash chain formed by the recursive application of a cryptographic hash function H. The security of the signatures and the time stamps depend on the security of the cryptographic hash function. It is assumed that the hash function used possesses all the fundamental security properties.

The usage of a hash chain records the audit logs of all the contract updates and the signatures. All contract updates starting from the initial formation of the contract till the final update are linked with each other using the hash chain. This ensures that if either of the party tries to change any document, then this change would reflect in all the following contract updates. The main attack scenario during contract management or archiving occurs if any party changes the contract update M_1 to M'_1 then this change is reflected in all the subsequent hash values in the hash chain. If that party tries to prove that the correct update is M'_1 but not M_1 after the contract has been established then the party has to make sure that the digest of the

final hash chain is the same as the original one on which both parties have signed before. This is hard for any party as it requires violation of the 2nd-preimage resistance property of the hash function used to form the hash chain and it is hard to violate this property for a secure hash function. In addition, the use of the hash chain and the time stamping of the finalised e-contract by a trusted TSA prevents fraudulent attempts by either of the parties on the other party such as not returning the e-contract update with its own signature by claiming falsely that it did send the signed e-contract.

Our contract management system does not explicitly include the request and agreement for change protocols between the parties as the multiple signing and dynamic update approaches of [2] to manage contract updates. In our architecture, these protocols are implicitly handled by the hash chain. In addition, there is no need to have a separate e-notary to act as a trusted third party to sign every contract update due to the use of the hash chain and the trusted TSA to time stamp the final hash chain. The absence of e-notary to sign every contract update ensures that our contract update protocol does not impose as much computational and communications burden on the contract management system as the schemes in [2]. In our architecture, every update requires the generation of two signatures and the verification of one signature.

5 Conclusion

In this paper, we have identified seven security goals for e-contracts and the mechanisms of an e-contracting system to achieve those security requirements. We have analysed the security of AEC collaborative platforms and their potential to be used for e-contracting. A new architecture for secure e-contracting which can be incorporated into existing AEC collaborative platforms is proposed. This new architecture is able to provide all the security requirements identified.

References

[1] National Institute of Standards and Technology (NIST) , Computer Systems Laboratory. Secure Hash Standard. Federal Information Processing Standards Publication(FIPS PUB) 180-2, August 2002.

[2] Samuil Angelov, Sven Till, and Paul Grefen. Dynamic and Secure B2B E-contract Update Management. In EC'05: Proceedings of the 6th ACM conference on Electronic commerce, pages 19{28. ACM Press, 2005.

[3] Arne Ansper, Ahto Buldas, Märt Saarepera, and Jan Willemson. Improving the availability of time-stamping services. In Y. Mu V. Varadharajan, editor, Information Security and Privacy : 6th Australasian Conference (ACISP), Lecture Notes in Computer Science, pages 360{375. Springer, 2001.

[4] Aconex Australasia. Aconex australasia - online information management solutions. The link is available at http://www.aconex.com/index.php?selectedSite=au. Last access date: 20th of October 2006.

[5] Paul Berning and Shaye Diveley-Coyne. E-commerce and the construction industry: The revolution is here, 2000. This paper is available at http://www.constructionweblinks.com/Resources/Industry_Reports_ _Newsletters/Oct_2_2000/e-commerce.htm. Last access date: 25th of October 2006.

[6] Ahto Buldas, Helger Lipmaa, and Berry Schoenmakers. Optimally effcient accountable time-stamping. In Hideki Imai and Yuliang Zheng, editors, PKC: International Workshop on Practice and Theory in Public Key Cryptography, volume 1751 of Lecture Notes in Computer Science, pages 293{305. Springer, 2000.

[7] Daniel Burgwinkel. Managing contractual relationships in virtual organizations with electronic contracting. In Luis M. Camarinha-Matos, editor, Collaborative Business Ecosystems and Virtual Enterprises, volume 213 of IFIP International Federation for Information Processing, chapter 12, pages 101{108. Springer, 1 edition, 2005.

[8] Citadon. Citadon Collaborative Workspaces. The link is available at http://citadoncw.citadon.com/support/CitadonCW/. Last access date: 31st of July 2006.

[9] Citadon. Security-Network Operations Security Policy. This document is obtained through a personal communication from the Director, Corporate Sales of Citadon.

[10] Autodesk Constructware. Autodesk Constructware - Features & Benefits. The link is available at http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=7104129. Last access date: 26th of July 2006.

[11] Autodesk Constructware. Causeway Behind the Profitability of the Construction Industry. The link is available at http://www.causeway.com. Last access date: 1st of August 2006.

[12] Tim Dierks and Christopher Allen. The TLS protocol version 1.0. Internet Request for Comment RFC 2246, Internet Engineering Task Force, January 1999. Proposed Standard.

[13] E-Builder. e-Builder - Construction Project Management. The link is available at http://www.e-builder.net. Last access date: 3rd of August 2006.

[14] E-Builder. TeamBinder Project Collaboration Software. The link is available at http://www.teambinder.com/teambinder/Home/. Last access date: 31st of July 2006.

[15] Kendall Freeman. Concluding Contracts by E-mail and the Use of Electronic Signatures. In-House Lawyer, Issue 132, July/August 2005. This article is available at the link http://www.kendallfreeman.com/publications/in-houselawyer.asp. Last Access date: 5th of May 2006.

[16] Alan Freier, Philip Karlton, and Paul Kocher. The SSL protocol version 3.0- internet draft, 1996. This Internet Draft is available at the link http://wp.netscape.com/eng/ssl3/ssl-toc.html.Last date of access: 22nd of October 2006.

[17] Dieter Gollmann. Computer Security, chapter Security Models. John Wiley & Sons, 1999.

[18] Mike Just. On the Temporal Authentication of Digital Data. PhD thesis, Carleton University, 1998.

[19] Mike Just. Some timestamping protocol failures. In Proceedings of the Symposium on Network and Distributed Systems Security (NDSS '98). Internet Society, March 1998. The paper is available at the link http://www.isoc.org/isoc/conferencesndss/98/just.pdf. Last access date:6th of April 2006.

[20] J.W.Carter and D.J.Harland. Contract Law in Australia. Butterworths, Sydney, 3rd edition, 1996.

[21] Konstantin Knorr and Susanne R□ohrig. Security Requirements of E-business Processes. In I3E, pages 73{86, 2001.

[22] William McGovern and Lary Lawrence. Contracts and Sales: Cases and Problems. Matthew Bender, Sydney, 1st edition, 1986.

[23] Arion Meier and Susanne Röhrig. Security Levels for Contracting Agents. In SEC, pages 495{506, 2002.

[24] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography, chapter Hash Functions and Data Integrity, pages 321{383. The CRC Press series on discrete mathematics and its applications. CRC Press, 1997.

[25] David L. Mills. Network time protocol (version 3) | specification, implementation and analysis. Internet draft standard RFC 1305, March 1992.

[26] National Institute of Standards and Technology. Federal Information Processing Standards (FIPS) PUB 186-2: Digital Signature Standard (DSS). pub-NIST, January 2000. The document is available at http://csrc.nist.gov/publications/fips/ fips186-2/fips186-2-change1.pdf. Last Access date: 3rd of April 2006.

[27] Susanne Röhrig and Konstantin Knorr. Security Analysis of Electronic Business Processes. Electronic Commerce Research, 4(1-2):59{81, 2004. [28] Security Technology Competence Centre (SETCCE). Trusted Electronic Archive. White Paper, September 2005. This document is available at http://www.setcce.si/eng/download/Trusted_Electronic_Archives_-_White_Paper.pdf.

[29] Richard Stim. License Your Invention: Sell Your Idea and Protect Your Rights with a Solid Contract \With CD", chapter Sample Agreement, pages 11{27. 2004.

[30] BIW Technologies. BIW Technologies Ltd - Services / Information Channel. The link is available at http://www.biwtech.com/services/ic.asp. Last access date: 2nd of August 2006.

[31] BIW Technologies. Web Based Collaboration Software & Online Document Management from Evoco - Specializing in Construction Project Management. The link is available at http://www.evoco.com. Last access date: 1st of August 2006. [32] Norman Walsh. A Technical Introduction to XML, 1998. The document is available at http://www.xml.com/pub/a/98/10/guide0.html. Last access date: 30th of October 2006.

[33] Paul Wilkison. Construction Collaboration Technologies- The extranet evolution. Taylor & Francis, 1st edition edition, 2005.